

Desenvolvimento Java Enterprise Edition usando JSF + Spring Framework e Hibernate

Uma aplicação Java EE da teoria a
prática
por Rodrigo Urubatan Ferreira Jardim

Sobre o Palestrante

- Rodrigo Urubatan - SCJP 1.4 e SCWCD
- Trabalha com arquitetura de sistemas J2EE e treinamento
- Já desenvolveu projetos utilizando as linguagens Delphi, C++, PHP, ASP, ColdFusion, Leather, Assembly, Perl, ...
- Trabalha com Java/J2EE a 4 anos e com desenvolvimento de sistemas a 9 anos
- Atualmente colabora com pequenas correções a alguns projetos Open Source como o G.U.J.2, Lombok e Velocity e faz parte da coordenação do RSJUG
- Já ministrou palestras em universidades (UCS, ULBRA, UNISC) e diversos eventos (Just Java, FISL, Seminário do RSJUG, Maratona 4 Java, Infosul) tutoriais para o RSJUG e já teve um artigo publicado na revista Mundo Java
- Atualmente trabalha como consultor na AdvancedIT, como gerente de tecnologia e qualidade na Tech Office IT, e ministra cursos e alguns pequenos projetos pela USI Informática
- É o principal desenvolvedor do projeto Spring-Annotation

Agenda

- Java 5
- Hibernate
 - Mapeamento Objeto Relacional
 - Configuração
 - Consultas
- Spring Framework
 - IoC/DI
 - Recursos do Framework
 - Porque usar
- JSF
 - Conceitos Básicos
 - Recursos

Hibernate

Hibernate é serviço de persistencia e consulta objeto/relacional poderoso e de alta performance. Hibernate permite o desenvolvimento de classes persistentes utilizando um idioma orientado a objetos – incluindo associações, herança, polimorfismo, composição, e coleções.

Hibernate

Hibernate permite a escrita de consultas no seu idioma próprio e portátil (HQL), bem como em SQL nativo, ou com uma API Orientada a objetos de Critérios e Exemplos.

Mapeamento Objeto Relacional

- Por que?
- Para que serve?
- Como fazer?
- Hands On!

Mapeamento O/R: Por que?

- Desenvolver qualquer tipo de sistemas ja não é uma tarefa simples.
- Precisar pensar de duas formas diferentes aumenta a complexidade.
- Pensar apenas de forma relacional nos faria perder a maior parte das vantagens do Java.

Mapeamento O/R: Para que serve?

- Mapear os objetos que precisam ser persistentes do sistema para a forma mais popular e performática de armazenamento de dados utilizada hoje.
- Permite pensar em todo o sistema na forma de objetos, e utilizar o banco de dados apenas como armazenamento de dados.
- Não é necessário conhecer duas linguagens diferentes para escrever um sistema.

Mapeamento O/R: Como Fazer?

- Criar um modelo de objetos que seja suficiente para trabalhar com os dados necessários para a aplicação.
- Tomar cuidado para realizar o mapeamento correto, seguindo os conceitos de orientação a objetos.
- Evitar de criar “objetos-tabela”

Hands On: O que pode ser mapeado

- Tipos primitivos do java
- Tipos wrapper
- Enums
- Classes persistentes
- Relacionamentos entre classes
- Componentes
- Coleções

Hands On: Abordagens para o mapeamento

- Anotações
 - Anotações compatíveis com a especificação da Java Persistence API (EJB3)
 - Um único lugar para saber onde as classes serão persistidas
 - Utiliza as informações de coleções genéricas
 - Menor quantidade de meta dados para o mesmo resultado
 - Extensões do Hibernate disponíveis onde a especificação de JPA não chegou.

Hands On: Anotações

- @Entity
- @Id
- @GeneratedValue
- @Embedded
- @Embeddable
- @EmbeddedId
- @Basic
- @Column
- @OneToMany
- @OneToOne
- @ManyToOne
- @ManyToMany
- @MapKey
- @MappedSuperClass
- Inheritance

Hands On: Enumerações

- DiscriminatorValue
- FetchType
- GenerationType
- InheritanceType
- CascadeType
- EnumType
- FlushModeType
- LockModeType

Hands On: Validação Integrada

- Com as classes mapeadas até o momento não temos nenhum tipo de validação
- Não há campos obrigatórios exceto o ID
- Isto aumenta a complexidade do resto do código, obrigando o código da aplicação a tratar isto explicitamente.

Hands On: Validação Integrada

- Hibernate Annotations possui uma extensão de validação
- Algumas das validações são refletidas no banco de dados gerado
- Podem ser utilizadas independente do resto do Hibernate quando desejado
- Possibilidade de criar validações customizadas

Hands On: Validação Integrada

- @Length(min=, max=)
- @Max(value=)
- @Min(value=)
- @NotNull
- @Past
- @Future
- @Email
- @Pattern(regex="regex p", flag=)
- @Range(min=, max=)
- @Size(min=, max=)
- @AssertFalse
- @AssertTrue
- @Valid

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda

rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Configuração

- O Hibernate para funcionar precisa de algumas informações
 - Como se conectar ao banco de dados
 - Qual o banco de dados que estamos utilizando
 - Quais as classes serão persistidas
 - Quais pacotes possuem configurações
- E possui diversas configurações opcionais que veremos a seguir

Configuração

- Suporte para conexão ao banco de dados
 - Utilizando um datasource configurado
 - Utilizando um datasource fornecido programaticamente
 - Utilizando um datasource fornecido via JNDI
 - Utilizando uma conexão fornecida manualmente

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda

rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Consultas

- Possibilidades para consultas
 - HQL
 - SQL Nativo
 - **Criteria**

Consultas: HQL

- Linguagem propria do Hibernate
- Similar a SQL, mas com extensões orientadas a objetos
- Facilitada a navegação nos relacionamentos por meio de propriedades
- Possibilita a utilização de poucos recursos propios do banco de dados

Consultas: SQL Nativo

- Aproveita o conhecimento já adquirido na empresa
- Possibilita a utilização de todo o poder do banco de dados
- Nos prende ao banco de dados utilizado
- Não utiliza todas as vantagens do mapeamento O/R

Consultas: Criterias API

- Linguagem totalmente orientada a objetos
- Consultas escritas em Java
- Facilidade para Query By Example
- Facilidade para escrever consultas dinâmicas
- Evita que o programador cometa alguns tipos de erros

Consultas: Design Pattern DAO

- Oculta do resto da aplicação qual a tecnologia de acesso a dados que esta sendo utilizada.
- Concentra em um unico lugar o conhecimento de como transportar um objeto de/para o banco.
- Facilita o tratamento de erros do banco de dados em um unico lugar.

Consultas: Generics

- Podemos utilizar os generics do Java 5 para escrever um código padrão para acesso a dados.
- A API do hibernate facilita o trabalho.
- A Criteria API nos fornece os recursos necessários para escrever código parametrizavel.
- Classes parametrizadas reduzem a complexidade do código.

Consultas: Implementação do Dao

- Precisa ser genérico como a interface
- Precisa suportar qualquer um dos objetos que criamos
- A API do Hibernate padrão já dá suporte a criação, atualização e deleção de objetos
- A API de Criteria dá suporte a consultas.
- Precisamos ter acesso a um SessionFactory do Hibernate.

Consultas: Session

- Session é equivalente a uma conexão ao banco de dados no Hibernate
- Session não é ThreadSafe
- Session é a nossa interface básica com a API do Hibernate

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda

rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Spring Framework

- J2EE deveria ser mais fácil de utilizar
- É melhor programar orientado a interfaces do que a classes. Spring reduz a complexidade de programar voltado a interfaces a quase zero.
- JavaBeans oferecem um ótimo meio de configurara aplicações.
- O design OO é mais importante do que qualquer tecnologia, como o J2EE.
- Exceções checadas são utilizadas em excesso no Java. Um framework não deveria te obrigar a tratar exceções, a não ser que fosse possível se recuperar delas.

O que é o Spring Framework

- Basicamente o Spring Framework é um conjunto de componentes reutilizáveis
 - Container de IoC
 - Framework AOP
 - MVC
 - Hierarquia de exceções
 - Facilidades para acesso a dados e utilização de frameworks OR
 - JMS
 - JMX
 - Scripting
 - ...

Inversão de Controle / Injeção de Dependencia

No início de 2004 Martin Fowler perguntou em um forum “quais aspectos do controle deste sistema estamos invertendo” e logo depois ele sugeriu que o nome do pattern/princípio fosse alterado, ou fosse um pouco mais “auto explicativo”, para uma maior explicação do que o Fowler chama de Injeção de dependencias, pode ser consultada a URL <http://martinfowler.com/articles/injection.html>

Bean Factory

- O que é e para que serve
- Bean Definition
- A classe para o Java Bean
 - Inicialização por construtor
 - Inicialização via factory method estático
 - Inicialização via factory method de instancia
- Identificadores para o Bean (id e name)
- Singleton ou não Singleton

Recursos avançados da BeanFactory

- Interfaces para o ciclo de vida
 - Inicialização/init-method
 - Finalização/destroy-method
- Sabendo quem você é
 - BeanNameAware
 - BeanFactoryAware
 - ApplicationContextAware
- Extendendo o framework
 - BeanFactoryPostProcessor
 - BeanPostProcessor

Integração do Spring Framework com o AspectJ

- A partir da versão 2.0 do spring, o framework AOP do spring framework esta bastante integrado ao AspectJ
- Esta integração tras basicamente dois grandes beneficios
 - Pode-se utilizar a linguagem de expressões do AspectJ para definir pointcuts e advices no Spring
 - Pode-se injetar dependencias em aspectos do AspectJ
- Uma ótima demonstração desta integração é a nova anotação `@Configurable`

Em que podemos utilizar AOP?

- Controle de transações.
- Logging para depuração.
- Injeção de dependências em beans que não são carregados no contexto do spring.
- Segurança.
- Tratamento de erros para o usuário (mas isto vamos deixar pra depois ...)

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda

rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Escrevendo a Lógica de negócio

- Por que criar classes para regra de negócios?
- O que colocar nestas classes?
- Como trabalhar com elas?
- Por que não utilizar direto os DAOs na classe que trata as requisições?

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda

rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Java Server Faces

A tecnologia JavaServer Faces é um framework server-side de componentes para interface com o usuário para aplicações WEB desenvolvidas utilizando a tecnologia Java. E inclui:

- Um conjunto de APIs para: Representar componentes UI e gerenciar o estado deles, tratar eventos e validação de entradas, definir a navegação de páginas, e dar suporte a internacionalização e acessibilidade.
- Uma biblioteca de tags JavaServer Pages (JSP) para expressar uma interface JavaServer Faces usando uma página JSP.

Managed Beans

- Beans gerenciados são os controladores do JSF, onde o código java é executado, semelhante ao Code Behind do .NET
- Estes beans são acessíveis via a Expression Language
- Botões e links nas páginas podem chamar métodos dos Managed Beans
- Podem ser criados eventos para alterações na interface com o usuário dentro de um Managed Bean
- Um Managed Bean é um Java Bean simples, sem nenhuma exigência a mais.

Converters

- Em uma requisição WEB tudo é texto.
- O JSF utiliza os converters para transformar texto no tipo de dados da propriedade do componente desejado.
- A implementação padrão já vem com alguns converters implementados.

Validators

- JSF por padrão não suporta validação no lado do cliente.
- A validação é feita por classes java chamadas Validators.
- Estes validators podem disponibilizar mensagens de erro para serem renderizadas nas páginas.

Componentes

- Os componentes JSF são a base para construção da interface com o usuário.
- Existem componentes padrão equivalentes a quase todas as tags HTML e diversos “helpers”
- Existem diversas bibliotecas de componentes disponíveis comerciais e open source.

Tags

- As tags JSF são apenas um meio de expressar os componentes em uma página JSP.
- Para cada componente, validator ou converter criado, deve-se criar também uma tag para poder utiliza-lo também em uma JSP, caso contrario este só sera acessivel via código java.

WYSIWYG

- Java Studio Creator
- JDeveloper
- Mãos a Obra!

Partes comuns das páginas

- Não existe uma API de templates padrão para o JSF
- Facelets vem ao socorro!

Escrevendo o código (MBeans)

- Métodos para ações de botões e links
- Métodos para eventos
- Utilizando os beans já definidos
- Configurando propriedades
- Alterando a UI a partir de código java

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda
rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Spring-Annotation

- Menos XML
- Melhor integração com JSF
- Melhor Integração com o Hibernate
- Menos pontos de manutenção na aplicação

Lets Play



Tech Office Soluções em Tecnologia da Informação Ltda
rodrigo@techoffice.com.br | www.techoffice.com.br | <http://blog.urubatan.com.br>

Executando a aplicação!

- Servidor de aplicações
- Banco de dados
- Deploy

Duvidas



Referencias

- <http://www.springframework.org>
- <http://www.hibernate.org>
- <http://myfaces.apache.org>
- <http://java.sun.com>
- <http://urubatan.com.br>
- <https://facelets.dev.java.net>
- <https://spring-annotation.dev.java.net>
- <http://blog.urubatan.com.br>